This paper was presented as part of the main technical program at IEEE INFOCOM 2011

Sensor Localization with Deterministic Accuracy Guarantee

Ryo Sugihara Rajesh K. Gupta
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, California 92093-0404
Email: {ryo, rgupta}@ucsd.edu

***Abstract*—Localizability of network or node is an important subproblem in sensor localization. While rigidity theory plays an important role in identifying several localizability conditions, major limitations are that the results are only applicable to generic frameworks and that the distance measurements need to be error-free. These limitations, in addition to the hardness of finding the node locations for a uniquely localizable graph, miss large portions of practical application scenarios that require sensor localization. In this paper, we describe a novel SDP-based formulation for analyzing node localizability and providing a deterministic upper bound of localization error. Unlike other optimization-based formulations for solving localization problem for the whole network, our formulation allows fine-grained evaluation on the localization accuracy per each node. Our formulation gives a sufficient condition for unique node localizability for any frameworks, i.e., not only for generic frameworks. Furthermore, we extend it for the case with measurement errors and for computing directional error bounds. We also design an iterative algorithm for large-scale networks and demonstrate the effectiveness by simulation experiments.**

I. Introduction

Location is a central concept in sensor networks for most application scenarios. In general, however, knowing the location of sensor nodes is not an easy task. GPS-based localization is usually too expensive both in terms of cost and energy for sensor networks and possible only in outdoor applications. More practical solution is to obtain distance information based on distance measurement between sensor nodes by using acoustic signals, signal strength, and so on; for more details on various ranging techniques see e.g., [1]. The problem of estimating the location of each sensor node from distance information between nodes is usually referred to as the sensor localization problem.

A number of studies on sensor localization have improved average accuracy of location estimation. What is less studied, however, is about the guarantee on accuracy. Depending on the application scenario, having the worst-case accuracy guarantee is as important as having good average performance. For example, in secure localization [2]–[5], where a prover makes a claim about its location against a group of remote verifiers and the verifiers check if the claim is valid or not, error of each verifier's location directly affects the reliability of a protocol. By providing a guarantee on accuracy, we can compute the region that the verifiers can assert the validity of the claim. Furthermore, by having a deterministic guarantee, we can adapt to applications that have hard requirements on reliability, which is often the case in security applications like this.

A. Limitations of Rigidity Theory

Localization accuracy is closely related to whether a node can be uniquely localized or not. The *unique localizability* in the sensor localization problem is studied mostly in the context of rigidity theory. In rigidity theory, to discuss rigidity as a property of graph instead of that of each framework[1], people usually consider *generic* frameworks, meaning the vertex coordinates are algebraically independent over the rationals, which roughly equals to "without degeneracy", i.e., no three points on the same line, no three lines go through the same point, etc. The reason is, even though frameworks are almost always rigid for a certain graph, some non-rigid frameworks can exist for the same graph. By assuming generic frameworks, we can rule out such cases as rare special cases. Specifically, it is shown that the probability of such cases for any reasonable probability distribution on node positions is zero [6]–[8].

However, we claim that sensor nodes are often deployed in an "unnatural" way. For example in two dimensional deployment, nodes are often placed at grid points, resulting in more than three nodes on a single line. Moreover in three dimensional deployment, nodes are very likely to be installed on the floor and ceiling, resulting in more than four nodes on a single plane[2]. Both of these are violations of genericness and it is irrelevant to discuss localizability based on the graph rigidity for these cases.

Figure 1(a) shows a simple example of such non-generic framework. When the node and two anchors are aligned on a single line, the node is uniquely localizable if the distance information is exact. However, a case like this is usually left out in an analysis of unique localizability based on rigidity theory. On the other hand, the node is not uniquely localizable in Figure 1(b) due to reflection, but the distance between two possible locations are not large. Depending on the application's requirement, even if a node is not uniquely localizable, it may suffice if all possible locations of the node are guaranteed to be within certain bound. Meanwhile in rigidity theory, uniquely

[1] i.e., an embedding of a graph in Euclidean space. Also called "realization" or "point formation."

[2] For instance, variable-length mounts are used to avoid this situation in [9].

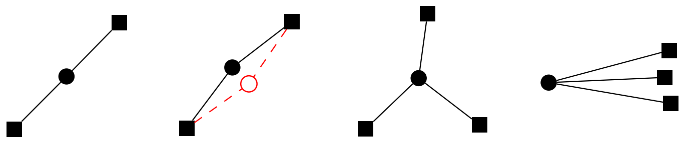

Fig. 1. Examples: Circles are nodes and squares are anchors: (a) Uniquely localizable case with two anchors; (b) Not uniquely localizable, but localization error determined by the distance between two possible locations is small; (c-1,2) Both nodes are uniquely localizable in error-free case, but the error will be larger in (c-2) under measurement errors.

localizable or not is the main question and the amount of possible localization error is not discussed.

Another limitation of rigidity theory arises when distance measurements are not accurate. Most results including unique localizability established through graph rigidity assume exact distance measurements. However, it is almost always the case in practice that distance measurements are not accurate, especially when they are obtained by each sensor node with limited hardware equipments. What is more problematic is, localization error due to inaccurate measurements is quite different depending on node configuration. For example, Figure 1(c-1) and (c-2) are two possible realizations of a graph. Although the node is uniquely localizable in both cases, the upper bounds of localization error under imprecise measurements are actually very different.

B. This Paper

In this paper we present semidefinite programming (SDP) formulations for computing an upper bound of localization error of each node in sensor network localization. The upper bounds are guaranteed to be correct, thus serving as a deterministic accuracy guarantee. While there are several papers on SDP formulations of sensor localization as we will review later, our formulation is unique in giving this strong accuracy guarantee.

The formulations are general in that they are not limited to generic frameworks. Nevertheless it provides some interesting connections with rigidity theory regarding unique localizability. We can also handle large scale networks through the iterative algorithm we design.

Our contributions in this paper are as follows:

- Presenting a new SDP formulation that gives deterministic accuracy guarantees for sensor node localization,
- Analyzing the error bound and unique localizability condition that are not restricted to generic frameworks,
- Extending the formulation for the case with inaccurate distance and for computing directional error bounds, and
- Designing an iterative algorithm appropriate for large-scale networks and demonstrating the effectiveness by simulations.

The rest of this paper is organized as follows. In Section II we present the SDP formulation for computing the localization error bound. Its analytical properties including the condition for unique localizability are discussed in Section III. In Section IV we extend the formulation for the case with measurement errors. For large-scale networks, we design an iterative algorithm in Section V. We show simulation results in Section VI. Related work is reviewed in Section VII and Section VIII concludes the paper.

II. Problem Formulation

In this section we give an SDP formulation for computing an upper bound of localization error, which is referred to as "error bound" hereafter. We first present a related SDP formulation for network localization problem by Biswas et al. [10] and describe how we extend that to evaluate a deterministic error bound for each node.

Throughout this section, we assume the distance information is exact. We also assume that network is realizable, i.e., there exists at least one feasible solution.

A. Preliminary: Network Localization

We are given a graph $G = (V_n \cup V_a, E_n \cup E_a)$ and consider its embedding in D-dimensional Euclidean space. V_n is the set of *nodes* whose locations are unknown and V_a is the set of *anchors* whose locations are known (denoted $a_k \in R^{D\times 1}$ for k-th anchor). E_n is the set of edges between nodes and, for each edge $(i,j) \in E_n$, we are given the distance d_{ij}. Similarly, E_a is the set of edges between a node and an anchor and the distance d_{jk} is given for each edge $(j,k) \in E_a$. The problem is to find the location (denoted $x_i \in R^{D\times 1}$ for i-th node) of all nodes in D-dimensional space such that all distance constraints

$$\begin{aligned} ||x_i - x_j||^2 &= d_{ji}^2, \forall (i,j) \in E_n, &(1)\\ ||x_j - a_k||^2 &= d_{jk}^2, \forall (j,k) \in E_a &(2) \end{aligned}$$

are satisfied, where $||\cdot||$ denotes the Euclidean norm.

This problem is formulated as an optimization problem. Let matrix $X = [x_1, x_2, ..., x_N] \in R^{D\times N}$, $Y \in R^{N\times N}$,

$$\begin{array}{ll} \text{find} & X, Y \\ \text{s.t.} & Y \bullet Q(e_{ij}) = d_{ij}^2 \ \forall (i,j) \in E_n \\ & \begin{bmatrix} I_D & X \\ X^T & Y \end{bmatrix} \bullet Q([a_k; -e_j]) = d_{jk}^2 \ \forall (j,k) \in E_a \\ & Y = X^T X, \end{array} \tag{3}$$

where $A \bullet B = \mathrm{Tr}(A^T B)$, $Q(v) = vv^T$, I_D is a D-dimensional identity matrix, $e_i \in R^{N\times 1}$ is a vector with all zeros except its i-th entry being one, and $e_{ij} = e_i - e_j$.

Since (3) is a nonconvex optimization problem, we obtain an SDP relaxation by replacing the quadratic equality constraint $Y = X^T X$ with a positive semidefiniteness constraint $Y \succeq X^T X$. Using Schur complement, we obtain the relaxed SDP problem as follows:

$$\begin{array}{ll} \text{find} & X, Y \\ \text{s.t.} & Y \bullet Q(e_{ij}) = d_{ij}^2 \ \forall (i,j) \in E_n \\ & \begin{bmatrix} I_D & X \\ X^T & Y \end{bmatrix} \bullet Q([a_k; -e_j]) = d_{jk}^2 \ \forall (j,k) \in E_a \\ & \begin{bmatrix} I_D & X \\ X^T & Y \end{bmatrix} \succeq \mathbf{0}. \end{array} \tag{4}$$

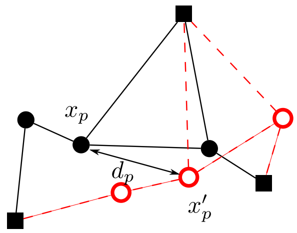

Fig. 2. Formulation idea: To calculate an upper bound of localization error of p-th node, consider two realizations of a graph and the distance between two locations x_p and x'_p.

B. Localization Error Bound and Node Localizability

Following the similar idea, we modify the formulation to maximize the distance between two corresponding nodes in two different realizations of a graph (Figure 2). Specifically, we consider two sets of node locations by having $X \in R^{D\times N}, X' \in R^{D\times N}, Y \in R^{2N\times 2N}$ as variables. For analyzing the localizability of p-th ($1 \leq p \leq N$) node, the objective is to maximize d_p^2, where d_p is the distance between p-th node in two realizations given by the following equation:

$$||x_p - x'_p||^2 = d_p^2 \tag{5}$$

In a matrix form, this is expressed as:

$$Z \bullet Q([\mathbf{0}; e_p; -e_p]) = d_p^2, \tag{6}$$

where

$$Z = \left[\begin{array}{c|cc} I_D & X & X' \\ \hline X^T & & \\ X'^T & \multicolumn{2}{c}{Y} \end{array}\right]. \tag{7}$$

Distance constraints between anchor and node are given by $||x_j - a_k||^2 = d_{jk}^2$ and $||x'_j - a_k||^2 = d_{jk}^2$, or alternatively,

$$\begin{aligned} Z \bullet Q([a_k; -e_j; \mathbf{0}]) &= d_{jk}^2, & (8)\\ Z \bullet Q([a_k; \mathbf{0}; -e_j]) &= d_{jk}^2. & (9)\end{aligned}$$

Similarly, distance constraints between two nodes are $||x_i - x_j||^2 = d_{ij}^2$ and $||x'_i - x'_j||^2 = d_{ij}^2$. Alternatively we have

$$\begin{aligned} Z \bullet Q([\mathbf{0}; e_{ij}; \mathbf{0}]) &= d_{ij}^2, & (10)\\ Z \bullet Q([\mathbf{0}; \mathbf{0}; e_{ij}]) &= d_{ij}^2. & (11)\end{aligned}$$

Finally, Y is an inner product matrix of $[X, X']$, i.e., $Y = [X, X']^T[X, X']$, but this is relaxed as $Y \succeq [X, X']^T[X, X']$. By Schur complement, this is equivalent to $Z \succeq \mathbf{0}$.

To wrap up, the SDP formulation for computing the error bound for p-th node is as follows:

$$\begin{array}{ll} \max\limits_{Z} & Z \bullet Q([\mathbf{0}; e_p; -e_p]) \\ \text{s.t.} & Z \bullet Q([a_k; -e_j; \mathbf{0}]) = d_{jk}^2 \quad \forall (j,k) \in E_a \\ & Z \bullet Q([a_k; \mathbf{0}; -e_j]) = d_{jk}^2 \quad \forall (j,k) \in E_a \\ & Z \bullet Q([\mathbf{0}; e_{ij}; \mathbf{0}]) = d_{ij}^2 \quad \forall (i,j) \in E_n \\ & Z \bullet Q([\mathbf{0}; \mathbf{0}; e_{ij}]) = d_{ij}^2 \quad \forall (i,j) \in E_n \\ & Z_{1:D,1:D} = I_D \\ & Z \succeq \mathbf{0} \end{array} \tag{12}$$

Let $\overline{d_p}$ denote the square root of the optimal value and d_p^* denote the true localization error. Since (12) is a relaxation and a maximization problem, we have $d_p^* \leq \overline{d_p}$, i.e., $\overline{d_p}$ is an upper bound of localization error for p-th node. If we have a feasible solution of the problem (3) and $\tilde{x_p}$ is the location of the p-th node in the solution, the true location x_p^* is guaranteed to satisfy $||x_p^* - \tilde{x_p}|| \leq \overline{d_p}$, i.e., x_p^* is in the circle centered at $\tilde{x_p}$ with radius $\overline{d_p}$. Moreover, since the optimal value of (12) is nonnegative, if we have $\overline{d_p} = 0$, we immediately obtain $d_p^* = 0$, i.e., unique localizability of the node. Therefore $\overline{d_p} = 0$ serves as a sufficient condition for unique node localizability.

However, there are two issues in practice. First, we need $\tilde{x_p}$ for the error bound to be useful, but it is not easy to compute a feasible solution for (3) to obtain $\tilde{x_p}$. Note that feasible solutions for (4) or (12) usually do not suffice, since the relaxation of the equality condition eliminates the constraint on dimension and as a result, the SDP formulation yields a solution in higher dimensional space [10], [11]. In fact finding a feasible solution for (3) is known to be NP-hard for error-free case [12]–[14], but we seek for a heuristic algorithm that works well in practice for the case with measurement errors, as we discuss later. The second issue is that, due to numerical errors, it is hard to show $\overline{d_p} = 0$ by solving (12) to establish the unique localizability. Instead, we will discuss a special case where $\overline{d_p} = 0$ is shown analytically by using dual problem.

C. Note on Quantitative Evaluation of Accuracy

It has been proposed that we can evaluate the accuracy of node location by the SDP formulation for network localization (4) [11], [15], [16]. Specifically, individual trace $Y_{jj} - ||x_j||^2 = 0$ is used as an indicator for node j to be uniquely localizable. While it gives a sufficient condition for unique localizability, it does not give quantitative information about localization error when a node is not uniquely localizable. Furthermore, it is not useful for the case with measurement errors, as is stated in [11]. Our formulation (12) can handle both of these cases and always give deterministic bounds on localization accuracy.

III. ANALYSIS OF LOCALIZATION ERROR BOUND AND UNIQUE LOCALIZABILITY

The dual of the SDP problem (12) is given as the following SDP problem:

$$\begin{array}{ll} \min\limits_{V,w,w',y,y'} & I_D \bullet V \\ & \quad + \sum\limits_{(j,k)\in E_a} (w_{jk} + w'_{jk}) d_{jk}^2 \\ & \quad + \sum\limits_{(i,j)\in E_n} (y_{ij} + y'_{ij}) d_{ij}^2 \\ \text{s.t.} & Q([\mathbf{0}; e_p; -e_p]) - \begin{bmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ & \quad - \sum\limits_{(j,k)\in E_a} \big(w_{jk} Q([a_k; -e_j; \mathbf{0}]) \\ & \qquad\qquad + w'_{jk} Q([a_k; \mathbf{0}; -e_j])\big) \\ & \quad - \sum\limits_{(i,j)\in E_n} \big(y_{ij} Q([\mathbf{0}; e_{ij}; \mathbf{0}]) \\ & \qquad\qquad + y'_{ij} Q([\mathbf{0}; \mathbf{0}; e_{ij}])\big) \preceq \mathbf{0} \end{array} \tag{13}$$

where $V \in R^{D\times D}$ and w, w', y, y' are scalars. Unlike the case in [11], there is no trivial feasible solution.

If there is a dual feasible solution such that the dual function becomes zero, we immediately know from duality theorem that there exists a primal feasible solution that makes the value zero, i.e., the node is uniquely localizable, since the primal function is the square of maximum localization error and thus nonnegative. This is also a case that the SDP relaxation solves the problem exactly. Unfortunately it is not very easy to find the conditions for such case in general, but we give results for a few simple cases.

First we start with the simplest setting where a node is directly connected with an anchor. In this case, we can analytically obtain the tight upper bound:

Claim 1. *If node p is only connected to one anchor k, the upper bound of localization error is $2d_{pk}$.*

Proof: We set $w_{pk} = w'_{pk} = 2, V = \mathbf{0}$ and all other w, w', y, y' to zero. Then the dual constraint becomes

$$\left[\begin{array}{c|cc} -4a_k a_k^T & 2a_k e_p^T & 2a_k e_p^T \\ \hline 2e_p a_k^T & -E_{pp} & -E_{pp} \\ 2e_p a_k^T & -E_{pp} & -E_{pp} \end{array}\right] \preceq \mathbf{0} \tag{14}$$

where E_{pp} is a $n\times n$ matrix filled with zero except the element at (p,p), which is one. Since the rank of this matrix is one and the sole nonzero eigenvalue is $-4||a_k||^2 - 2 < 0$, the constraint is satisfied.

The dual value for this solution is $4d_{pk}^2$, and from the duality theorem, the primal optimal value is equal to or less than this. Since the primal optimal value is the square of maximum localization error, we have the bound $\overline{d_p} \le 2d_{pk}$. ∎

Second we give a unique localizability result:

Claim 2. *For node p, if there exists $\{c_k\}$ $(k = 1, ..., |V_a|)$ such that $\sum_k c_k \neq 0, \sum_{k:(p,k)\in E_a} c_k a_k = \mathbf{0}$, then it is uniquely localizable.*

Proof: Since $\sum c_k \neq 0$, there exists a constant α such that $\alpha \sum c_k = -2$. Using α, we set $w_{pk} = w'_{pk} = -\alpha c_k$ for each $k : (p,k) \in E_a$ and $V = 2\alpha \sum_{k:(p,k)\in E_a} c_k a_k a_k^T$. All other w, w', y, y' are set to zero. Then the dual constraint is

$$\left[\begin{array}{c|cc} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & -E_{pp} & -E_{pp} \\ \mathbf{0} & -E_{pp} & -E_{pp} \end{array}\right] \preceq \mathbf{0}. \tag{15}$$

This inequality is satisfied (eigenvalues are -2 and 0) and the value of the dual objective function is

$$\begin{aligned} & 2\alpha \sum_{k:(p,k)\in E_a} c_k ||a_k||^2 - 2\alpha \sum_{k:(p,k)\in E_a} c_k d_{pk}^2 \\ = \; & 2\alpha \sum_{k:(p,k)\in E_a} c_k (||a_k||^2 - d_{pk}^2) \\ = \; & 2\alpha \left(2x_p^T \sum_{k:(p,k)\in E_a} c_k a_k - ||x_p||^2 \sum_{k:(p,k)\in E_a} c_k \right) \\ = \; & 4||x_p||^2 \end{aligned}$$

where the transformation between the second and third lines is by the law of cosines.

This value depends on $||x_p||$, the distance between the origin O and node p. However, the unique localizability (or maximum localization error) of node p should not be affected by the choice of O. In other words, duality theorem always holds for any choice of O and thus the primal optimal value needs to be equal to or less than the minimum possible value of the dual objective function, which is zero when $x_p = \mathbf{0}$. Since the primal value is nonnegative, it is also zero and node p is uniquely localizable. ∎

This claim may look trivial but has several important implications. First is the unique localizability in a trilateration graph [13]. When a node is directly connected to $(D+1)$ or more anchors, it is uniquely localizable since there always exists satisfying $\{c_k\}$. Since we can consider such uniquely localizable node as an anchor, we can iteratively reformulate the problem to enumerate all nodes on a trilateration graph.

Second point is about the case when a node is only connected with less than $(D+1)$ anchors. When the node is directly connected to $m(< D+1)$ anchors, for satisfying $\{c_k\}$ to exist, O must be chosen such that the dimension of the space spanned by $\{a_k\}$ is less than or equal to m. This means, when $m = 2$, O must be on the same line with the two anchors, and when $m = 3$, O must be on the same plane with the three anchors. Since $x_p = \mathbf{0}$ is necessary to make the dual solution zero, node p needs to satisfy the same constraint as O. This case shows a unique localizability of non-generic framework shown in Figure 1(a), which is a case explicitly avoided in localizability analysis based on graph rigidity theory.

IV. Extensions

In this section we give two extensions for the SDP problem (12). One is for the case when the distance information is not exact, and the other is for computing the error bound for a particular direction or a directional range.

A. Case with Measurement Errors

In the case with measurement errors, the assumption is that we are given with an interval that is guaranteed to contain the true distance. Other than that, we do not assume any statistical properties of error. We assume such guarantee is given for each measurement technique as a part of the specification.

1) Error Model: Biswas et al. [10] presented two error models for distance measurements: stochastic model and interval-based model. In the stochastic model each measurement follows Gaussian distribution around the true value. While this model is more amenable for analytical approaches such as maximum likelihood estimation, it is often hard to justify the Gaussian noise model. For example, when any ranging techniques based on signal propagation time (e.g., ultrasound) are used, erroneous measurements due to multipath effect are always larger than the true value obtained from signal via the direct path. For this reason we use the interval-based model that we described above.

2) Formulation: Given a distance interval $[\underline{d}, \overline{d}]$ that satisfies $\underline{d} \leq d \leq \overline{d}$ for unknown true distance d, the network localization problem that corresponds to (3) is as follows:

$$\begin{array}{ll} \text{find} & X, Y \\ \text{s.t.} & \underline{d_{ij}}^2 \leq Y \bullet Q(e_{ij}) \leq \overline{d_{ij}}^2 \ \forall (i,j) \in E_n \\ & \underline{d_{jk}}^2 \leq \begin{bmatrix} I_D & X \\ X^T & Y \end{bmatrix} \bullet Q([a_k; -e_j]) \leq \overline{d_{jk}}^2 \\ & \qquad\qquad\qquad\qquad\qquad \forall (j,k) \in E_a \\ & Y = X^T X. \end{array} \tag{16}$$

We also have an SDP relaxation of this problem by replacing $Y = X^T X$ with $Y \succeq X^T X$.

Similarly we replace each of the equality constraints in (12) by inequalities and obtain a modified SDP problem for computing the localization error bound as follows:

$$\begin{array}{ll} \max\limits_{Z} & Z \bullet Q([\mathbf{0}; e_p; -e_p]) \\ \text{s.t.} & \underline{d_{jk}}^2 \leq Z \bullet Q([a_k; -e_j; \mathbf{0}]) \leq \overline{d_{jk}}^2 \ \forall (j,k) \in E_a \\ & \underline{d_{jk}}^2 \leq Z \bullet Q([a_k; \mathbf{0}; -e_j]) \leq \overline{d_{jk}}^2 \ \forall (j,k) \in E_a \\ & \underline{d_{ij}}^2 \leq Z \bullet Q([\mathbf{0}; e_{ij}; \mathbf{0}]) \leq \overline{d_{ij}}^2 \ \forall (i,j) \in E_n \\ & \underline{d_{ij}}^2 \leq Z \bullet Q([\mathbf{0}; \mathbf{0}; e_{ij}]) \leq \overline{d_{ij}}^2 \ \forall (i,j) \in E_n \\ & Z_{1:D,1:D} = I_D \\ & Z \succeq \mathbf{0} \end{array} \tag{17}$$

B. Directional Error Bounds

By formulating the problem in a slightly different way, we can calculate the error bound for a specific direction or the directional error characteristics. They are useful in several cases; for example to choose a pair of nodes to add a distance measurement to reduce the error bound of certain node.

Here we discuss the two dimensional ($D = 2$) case only, but it is easy to extend it to $D = 3$ case. The directional constraint can be expressed by $(x_p - x'_p)^T r_\theta = ||x_p - x'_p||$, where $r_\theta = [\cos\theta; \sin\theta]$. Using Z we have

$$Z \bullet \left[\begin{array}{c|cc} \mathbf{0} & r_\theta e_p^T & -r_\theta e_p^T \\ \hline e_p r_\theta^T & & \\ -e_p r_\theta^T & \multicolumn{2}{c}{\mathbf{0}} \end{array} \right] = 2 d_p, \tag{18}$$

which is added to the constraint and the objective is changed to maximize d_p in (12) or (17). Note that θ is a given constant and not a variable.

To calculate the accuracy guarantee for a small angle range $\theta_{min} \leq \theta \leq \theta_{max}$ $(\theta_{max} - \theta_{min} \leq \pi/2)$, we replace (18) with the following inequality:

$$Z \bullet \left[\begin{array}{c|cc} \mathbf{0} & r_\alpha e_p^T & -r_\alpha e_p^T \\ \hline e_p r_\alpha^T & & \\ -e_p r_\alpha^T & \multicolumn{2}{c}{\mathbf{0}} \end{array} \right] \geq 2 d_p \cos\beta, \tag{19}$$

where $\alpha = (\theta_{min} + \theta_{max})/2$ and $\beta = (\theta_{max} - \theta_{min})/2$.

V. Iterative Algorithm for Large Networks

The SDP formulations (12) and (17) give an upper bound of localization error, but the size of the problem tends to be large. For instance, (12) has $2N(D + 2N)$ variables with $2|E_n + E_a| + 2$ constraints, while the ordinary network localization (4) has $N(D + N)$ variables with $|E_n + E_a| + 1$ constraints, where N is the number of nodes. As a result, it is hard to solve it for relatively small (~ 50 nodes) network due to memory limitation.

To deal with large problems, we design an algorithm that iteratively solves a small SDP problem for a subgraph of the original graph. Since these small problems are relaxations of the original problem, every intermediate solution also gives an upper bound, albeit looser than the solution of the original problem. In this section we focus on the case with measurement errors, which generalizes the error-free case.

A. Techniques for Scalability

1) Safe Elimination: Upon computing the error bound for a node, some nodes may not affect the bound at all. We call it a *safe elimination* when we can eliminate nodes or edges without affecting the error bound. There are some conditions that we can safely eliminate nodes and all incident edges from computation as follows:

Proposition 1. *If all paths from node v to node w contain at least one anchor, w and all incident edges can be safely eliminated without affecting the localization error of v.*

Proposition 2. *Node w can be safely eliminated if for all $a \in V_a$, all paths from w to anchor a contain node v.*

Unfortunately, except for very artificial graphs, there are not so many nodes safely eliminated by these conditions. In practice we need to reduce the problem size further and the notion of *pseudo-anchors* is useful for that purpose.

2) Pseudo-anchors: In the error-free case, a node can be uniquely localized when it has $(D + 1)$ directly connected anchors. Then the node can work as an anchor itself and this is how we can efficiently localize a trilateration graph.

In the case with measurement errors, however, localization error remains even for the nodes that were uniquely localizable in the error-free case. Thus they cannot be used for establishing unique localizability of other nodes. Nonetheless, since their errors are smaller than those for other nodes that are not uniquely localizable even without errors, their contribution for reducing the error tends to be large. Therefore we use these nodes that are uniquely localizable in the error-free case as "pseudo-anchors."

We can incorporate the computed error bounds of pseudo-anchors into the error calculation of a new node. This is done by adding $||\tilde{x}_p - x_p||^2 \leq \overline{d_p}^2$ and $||\tilde{x}_p - x'_p||^2 \leq \overline{d_p}^2$ to the problem, where $\tilde{x}_p$ is the location obtained from a feasible solution of (16).

One caveat about the use of pseudo-anchors is that obtaining $\tilde{x}_p$ is not easy in general, which we discuss in detail in the next subsection. Here we describe that we can use different subgraphs for obtaining a feasible solution and for calculating an upper bound of localization error. In particular, we have the following proposition:

Proposition 3. *Let $\overline{d_p}$ be an upper bound of localization error of node p obtained by solving (17) for $G' = (V'_n \cup V'_a, E'_n \cup E'_a)$*

s.t. $p \in V_n', G' \subseteq G$. *If* $\tilde{x}_p$ *is a feasible solution of (16) for* G'' *s.t.* $G' \subseteq G'' \subseteq G$, *true location* x_p *satisfies* $||x_p - \tilde{x}_p|| \leq \overline{d_p}$.

This is clear from that we can see the problem (16) on G' as a relaxation of the same problem on G''. Note that this guarantee does not hold in general when $G' \not\subseteq G''$.

B. Finding a Feasible Solution

The SDP formulation (17) provides an upper bound of localization error, but not the node location itself. To find the node location, we need to solve (16), a nonconvex optimization problem. In fact, even if we can determine the unique localizability of a graph, finding a realization for the graph is NP-hard in general [12], except for certain subsets of uniquely localizable graphs such as trilateration graphs [13] and universally rigid graphs [17]. Despite the hardness, we have a practical necessity for finding a feasible solution. The error bound of a node is not very meaningful without its location, and moreover, we need the location of a node to use it as a pseudo-anchor.

When we consider the case with measurement errors, the feasible space (i.e., the set of solutions that satisfy all the constraints) of the network localization problem (16) is larger than that of the error-free case (3). This suggests it is easier in practice to find a solution for (16). Although we do not have a guarantee, we can use local search to find a feasible solution.

We use a method based on gradient descent until reaching a point that satisfies all constraints. Using gradient descent for SDP solutions to improve the quality of estimation has been proposed in several works [10], [18], [19]. The difference from normal gradient descent is that we only consider the gradient of the violated constraints and not of the satisfied constraints. Note that we do not try to find the "center" in the feasible set: we just want one feasible point, since for any feasible point, it is guaranteed that the true location is in the circle with the error upper bound as the radius from that point.

In each iteration, x_i is updated based on the following rule:

$$x_i \leftarrow x_i + \eta\Big(\sum_{(i,j)\in E_n} g(x_i, x_j, \underline{d_{ij}}, \overline{d_{ij}}) + \sum_{(i,k)\in E_a} g(x_i, a_k, \underline{d_{ik}}, \overline{d_{ik}}) + g(x_i, \tilde{x}_i, -\infty, \overline{d_i})\Big) \tag{20}$$

where η is a positive constant and

$$g(x, y, \underline{d}, \overline{d}) = \begin{cases} \frac{x-y}{||x-y||} & \text{if } ||x-y|| < \underline{d} \\ -\frac{x-y}{||x-y||} & \text{if } ||x-y|| > \overline{d} \\ 0 & \text{Otherwise} \end{cases} . \tag{21}$$

The last term applies when node i already has a feasible solution $\tilde{x}_i$ and error bound $\overline{d_i}$.

Algorithm 1 summarizes the procedures for finding a feasible solution. In line 3, a node i is randomly chosen from the nodes not in $\tilde{V}_n$, the set of nodes with feasible solution. In line 4 and 5, we add all anchors and nodes within k-hop distance from node i, where k is a constant and is set to $k = 3$ in the simulation experiments. In line 6, we use maximum variance unfolding (MVU) with matrix factorization proposed by Weinberger et al. [19] to efficiently get an initial point for running the modified gradient descent. After obtaining a feasible solution for the subgraph, we try to merge it with the feasible solution for $\tilde{V}_n$ to see if there is a consistent solution (lines 8-13).

Algorithm 1 Finding a feasible solution

1: $\tilde{V}_n \leftarrow \emptyset$ ▷ Set of nodes with feasible solution
2: **while** $|\tilde{V}_n| < N$ **do**
3: Randomly choose node $i \in V_n - \tilde{V}_n$
4: $V_{i,n} \leftarrow \{v | v \in V_n, H(i,v) \leq k\}$ ▷ H : hop distance
5: $V_{i,a} \leftarrow \{a | a \in V_a, H(i,a) \leq k\}$
6: Run MVU on induced subgraph formed by $V_{i,n} \cup V_{i,a}$
7: Run modified gradient descent (using rule (20))
8: **if** feasible solution is found **then**
9: Run modified gradient descent on $\tilde{V}_n \cup V_{i,n}$
10: **if** feasible solution is found **then**
11: $\tilde{V}_n \leftarrow \tilde{V}_n \cup V_{i,n}$
12: **end if**
13: **end if**
14: **end while**

This algorithm may fail since the problem in general is NP-hard. However, even if the algorithm fails to find a feasible solution for the whole network, it is likely that we can still find feasible solutions for a subgraph. Therefore, assuming we have a feasible solution for subgraph $G'' \subset G$, we can calculate the error bound $\overline{d_p'}$ for subgraph $G' \subseteq G''$. Then from Proposition 3, for the pair of error bound $\overline{d_p'}$ and feasible solution $\tilde{x}_p'$, we have a guarantee that $||\tilde{x}_p' - x_p^*|| \leq \overline{d_p'}$.

C. Algorithm for Error Bound Calculation

Algorithm 2 shows how we compute the error bound for each node. In the initialization (lines 1-4), for each node we choose up to M_{ini} neighbor nodes (set to $M_{ini} = 10$ in the experiments) by first adding anchors and then pseudo-anchors with small error bounds. For node i, $N_a(i)$ denotes the set of anchors in the neighbor and $N_n(i)$ denotes that of nodes.

The main loop from line 5 is iterated for whole set of nodes V_n so that we can obtain rough error bounds for all nodes in a reasonable amount of time and then more precise ones afterwards. The inner iteration from line 7 starts with choosing a node to compute the error bound. The idea of how we choose a node is to start with the node with the least error bound so that other nodes can possibly benefit from that. Specifically, we sort all remaining nodes with multiple criteria in lines 9-13 and choose the best one.

For the chosen node i, we add M_{add} anchors/nodes (set to $M_{add} = 3$ in the experiments) (line 15). Ideally we want to add the anchors or nodes that contribute the most to reducing the error bound of node i. However, under measurement errors, it is hard to tell which node is the best for reducing the error bound. Here we choose the one with the minimum weighted error bounds. We use hop distance from i as the weight so that we can add more near nodes rather than far nodes. Then

Algorithm 2 Computing the error bound

```
1: for all i ∈ V_n do                              ▷ Prepare initial nodeset
2:     V_i ← Up to M_ini anchors in k ∈ N_a(i) with least d_ik
3:     V_i ← V_i ∪ Up to (M_ini − |V_i|) nodes in j ∈ N_n(i)
               with least d̄_j  ▷ Choose "good" pseudo-anchors
4: end for
5: while forever do                                ▷ Main loop
6:     V_iter ← V_n
7:     while V_iter ≠ ∅ do                     ▷ Iteration for each node
8:         i ← First node after sorting all nodes in V_iter
               by the following (in the order of precedence):
9:             min d̄_i,                          ▷ Error bound of itself
10:            max |N_a(i)|,                     ▷ #anchor in neighbor
11:            min avg_{k∈N_a(i)} (d̄_ik − d̲_ik),
12:            max |{j | j ∈ N_n(i), d̄_j < ∞}|,
13:            min avg{d̄_j | j ∈ N_n(i), d̄_j < ∞}|
14:        V_iter ← V_iter \ i
15:        V_i ← V_i ∪ Up to M_add nodes j ∈ N_n(V_i)
                   with least H(i, j) d̄_j
16:        G_i ← induced subgraph of G formed by V_i
17:        Update d̄_i by solving SDP problem (17) for G_i
18:    end while
19: end while
```

finally we consider the induced subgraph for the node set V_i and solve the SDP problem (17) to update the error bound $\overline{d_i}$ (line 16-17).

Convergence to $\overline{d_p}$ obtained for whole graph is guaranteed by the fact that each computation eventually contains all the nodes in the network. This heuristic algorithm is intended as a practical solution for the case that it is impossible due to a memory limitation etc. to solve the problem for the whole network. In practice, it takes more time for each iteration as the problem size gets bigger, so we stop the computation after certain number of iterations and use the value at the time. Since the problem for each subgraph is a relaxation of that for the whole graph, the error bound from intermediate problem serves as a conservative error bound.

VI. Evaluation

We implemented the SDP formulation (17) as well as Algorithm 1 and 2 to calculate an upper bound of localization error. We use Matlab and SeDuMi [20] with CVX [21] for solving SDP problems. We also use the Matlab implementation of maximum variance unfolding by Weinberger [22].

A. Small Graphs

First we compute the error bound for several small graphs to see how the measurement error affect the bound. We use a proportional noise represented by noise factor δ. For true distance d^*, we randomly generate a distance measurement d such that $(1-\delta)d \leq d^* \leq (1+\delta)d$ is satisfied. Note that this noise model is only for experiments and the SDP formulation itself can handle any noise model as long as an interval that is guaranteed to contain d^* is given.

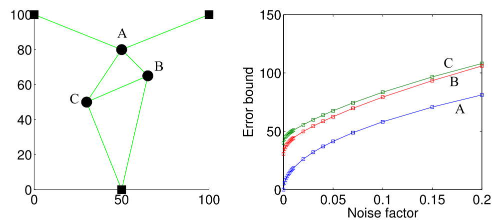

Fig. 3. Example 1: Case that RRT-3Beacon cannot identify the unique localizability of node A; Squares are anchors, circles are nodes.

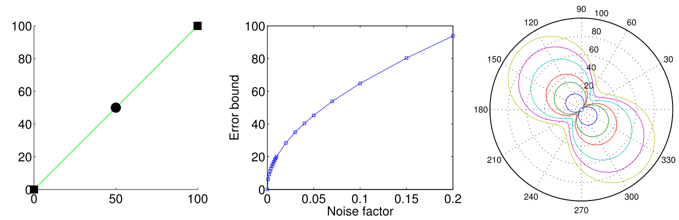

Fig. 4. Example 2: Case that unique localizability is established by two anchors due to non-generic configuration. Directional error bounds are shown for noise factor 0.01, 0.03, 0.05, 0.10, 0.15, 0.20.

Figure 3 shows a graph with three nodes and three anchors. This graph appears in [23] as an example that the sufficient condition for unique node localizability called RRT-3Beacon fails for node A, while it is actually uniquely localizable. The graph on the right shows the error bounds obtained by solving the SDP problem (17) for various noise factors from 0 to 0.2. As the graph shows, the error bound for node A approaches zero when the noise factor is close to zero, implying the unique localizability.

Figure 4 shows an example of non-generic framework with one node and two anchors on a single line. As is expected, error bound approaches zero for the cases with small noise factors. On the right the directional error bounds are shown. We divided the range $[0, \pi]$ into 100 smaller ranges and solved the SDP problem for each case. The error bounds for $[\pi, 2\pi]$ are obtained from symmetry. The results show that we have more uncertainty in the direction orthogonal to the edges.

Figure 5 shows two frameworks that have the same graph but with different anchor locations. In both cases the node is localizable and the error bound certainly approaches zero when the noise factor is close to zero. However, the growth of error bound is very different in two cases. The reason is clear from the directional error bounds, which show the uncertainty on $\theta = \pi/4$ direction is large in the second case.

B. Large Graphs

Next we evaluate the performance of Algorithms 1 and 2 with large graphs. For given number of nodes N, number of anchors K, and range r, nodes and anchors are randomly scattered in $[0, 1] \times [0, 1]$ region and nodes (or node and anchor) within r are connected with an edge.

First we evaluated the performance of Algorithm 1 to find a feasible solution of the network localization problem (16). Figure 6 shows the success rate for various parameters. The

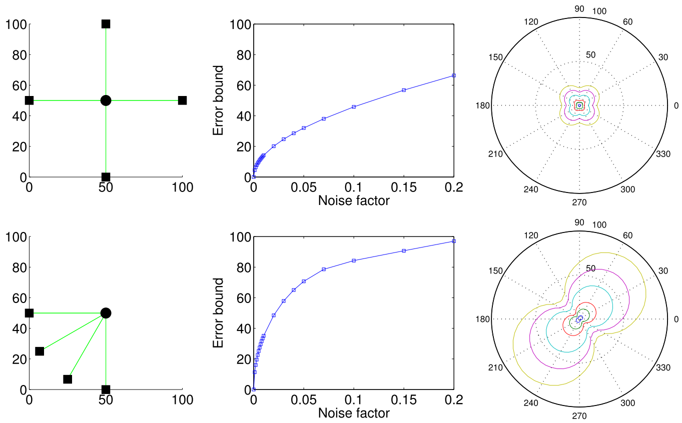

Fig. 5. Example 3: Two graphs have the same topology and the same distance, yet the error bounds are different under measurement errors due to anchors' locations. Directional error bounds are shown for noise factor 0.01, 0.03, 0.05, 0.10, 0.15, 0.20.

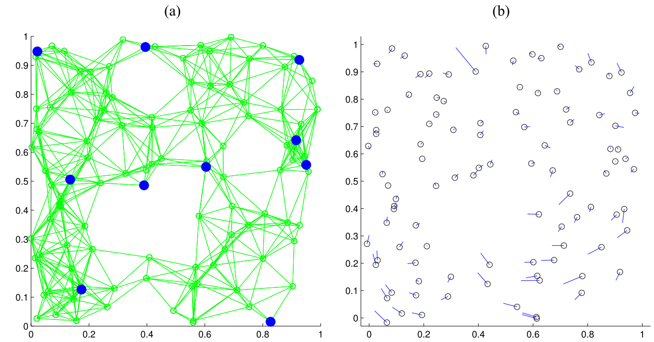

Fig. 7. Example of large network: (a) Ground truth: 100 nodes, 10 anchors, range $r = 0.2$, large dots are anchors; (b) An example of feasible solution: Under noise factor $\delta = 0.2$. For each node, estimated location is shown by a circle with the line representing the difference from the true location.

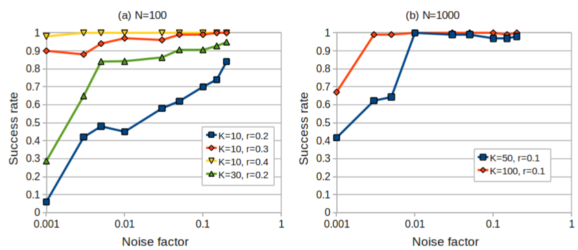

Fig. 6. Success rate of finding a feasible solution for the network localization problem (16): (a) Number of nodes $N = 100$ with various number of anchors K and range r; (b) $N = 1000$ case.

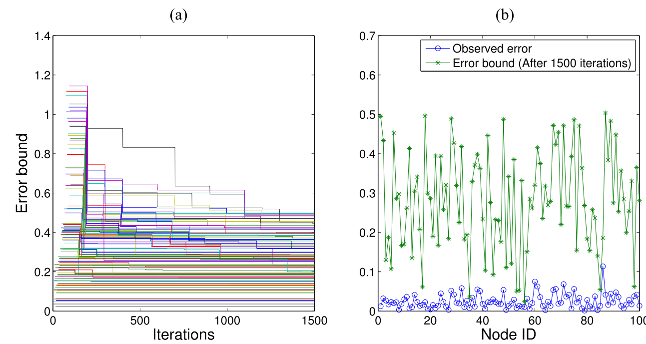

Fig. 8. Computation results for the network shown in Figure 7: (a) Progress of error upper bound for the first 1500 iterations; (b) Comparison of observed error and error bound for each node.

rate is very close to 100% for dense graphs with large noise factors, and decreases when the network is sparse, the anchors are few, or noise factor is small.

Figure 7(a) is an example of sensor deployments with 100 nodes, 10 anchors, and range $r = 0.2$. Figure 7(b) shows an example of feasible solution when the noise factor $\delta = 0.2$, with the differences from the ground truth as lines. Figure 8(a) shows the progress of error bounds computed by Algorithm 2 for the same network with the same noise factor. The graph shows the error bounds for the first 1500 iterations, which means 15 iterations for each of 100 nodes. For many of the nodes the error bound decreases the most in the second iteration (101st-200th iterations). Figure 8(b) shows the comparison of the observed error (shown in Figure 7(b)) and the error bound for each node. For all nodes the error bounds are larger than the observed errors.

VII. Related Work

Sensor localization problem consists of two parts: localizability problem and localization problem, where the former asks the existence of a framework that satisfies all distance constraints and the latter asks for the locations of each node for a localizable graph.

Rigidity theory is mainly for the localizability problem. Based on classical results [24], [25], Hendrickson [6] showed several necessary conditions for generic global rigidity, i.e., all generic frameworks are unique. Jackson and Jordan [26] showed that 3-connectedness and redundant rigidity are an equivalent condition for generic global rigidity in two dimensional space. For three dimensional case, such equivalent condition has been conjectured by Connelly [27] and proved by Gortler et al. [28]. Several subclasses of generically globally rigid graphs have been studied: Eren et al. [13] identified trilateration graphs and Zhu et al. [17] showed generically universally rigid graphs. As we have discussed earlier, almost all of these results are for generic frameworks.

In rigidity theory, node localizability has not been studied as much as network localizability. Goldenberg et al. [23] showed that having three node-disjoint paths to three distinct beacons is a necessary condition for unique node localizability and that RRT-3Beacon (a node belongs to globally rigid subgraph and contains three beacons) is a sufficient condition. Tighter conditions are found by Eren et al. [29] by introducing *implicit edges* and then further improved by Yang and Liu [30].

Another approach for sensor localization problem is through formulation as optimization problems. Optimization approach is mostly concerned with the localization problem. Network localization problem can be formulated as a nonconvex opti-

mization problem and its relaxation to SDP problem has been studied [10], [11], [15]–[19]. As we have discussed in earlier sections, our work is built upon these to specialize in providing deterministic upper bounds on localization error of each node.

Several work has analytical treatments of noisy distance measurements. Biswas et al. [10] presented SDP formulations for stochastic and interval-based noise models. They provided a detailed analysis only on the stochastic error case, and we have presented one for the latter case in this paper. In another paper [18], they also presented techniques for refining the localization results by regularization and gradient descent. Jian et al. [31] proposed a way to filter out erroneous distance information by analyzing embeddability of a graph by rigidity theory. However, their method is limited in the sense that all non-erroneous distances are assumed to be exact. Moore et al. [32] considered noisy range measurement cases and presented an algorithm to avoid flip ambiguities. Our SDP formulation automatically takes care of such ambiguities; when a node has a flip ambiguity, the error bound will be large. Liu and Zhang [33] focus on error-control mechanism based on characterizing the node uncertainties, but it does not give any deterministic guarantee.

VIII. Conclusion

We have presented an SDP formulation to quantitatively evaluate the deterministic upper bound of localization error of each node in sensor localization problem. Our formulation is general in multitude of ways: it is not limited to generic frameworks, it is easily extended to the case with measurement errors and also for computing directional error bounds. We also designed an iterative algorithm so that we can solve the problem for large-scale networks without destroying the correctness of the upper bound, along with another heuristic algorithm for finding a feasible solution for network localization problem. Simulation results demonstrate that the formulation is effective in handling the cases that have been eliminated as exceptions in rigidity-based studies, and also useful in practice for knowing the worst-case localization error for each node.

Acknowledgment

This work was supported in part by UCSD/LANL Engineering Institute and by NSF under grant numbers CNS-0932360, CCF-0702792, and SRS-0820034. The first author thanks Muhammad Abdullah Adnan for fruitful discussions.

References

[1] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 51, no. 10, pp. 2529–2553, 2007.

[2] S. Brands and D. Chaum, "Distance-bounding protocols," in *EUROCRYPT*, vol. 765. Springer, 1994, pp. 344–359.

[3] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proceedings of the 2nd ACM workshop on Wireless security*, 2003, pp. 1–10.

[4] S. Capkun, M. Cagalj, and M. Srivastava, "Secure Localization with Hidden and Mobile Base Stations," in *INFOCOM*, 2006.

[5] N. Chandran, V. Goyal, R. Moriarty, and R. Ostrovsky, "Position Based Cryptography," in *CRYPTO*, 2009, pp. 391–407.

[6] B. Hendrickson, "Conditions for unique graph realizations," *SIAM Journal on Computing*, vol. 21, no. 1, pp. 65–84, 1992.

[7] B. Roth, "Rigid and flexible frameworks," *American Mathematical Monthly*, vol. 88, no. 1, pp. 6–21, 1981.

[8] L. Lovasz and Y. Yemini, "On generic rigidity in the plane," *SIAM Journal on Algebraic and Discrete Methods*, vol. 3, no. 1, pp. 91–98, 1982.

[9] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-assisted localization in wireless sensor networks," in *INFOCOM*, 2005, pp. 172–183.

[10] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Transactions on Sensor Networks*, vol. 2, no. 2, pp. 188–220, 2006.

[11] A. M.-C. So and Y. Ye, "Theory of semidefinite programming for Sensor Network Localization," *Mathematical Programming*, vol. 109, no. 2-3, pp. 367–384, 2006.

[12] J. Saxe, "Embeddability of weighted graphs in k-space is strongly NP-hard," in *Proc. 17th Allerton Conf. Commun. Control Comput*, 1979, pp. 480–489.

[13] T. Eren, O. Goldenberg, W. Whiteley, Y. Yang, A. Morse, B. Anderson, and P. Belhumeur, "Rigidity, computation, and randomization in network localization," in *INFOCOM*, 2004, pp. 2673–2684.

[14] J. Aspnes, D. Goldenberg, and Y. Yang, "On the computational complexity of sensor network localization," in *Algorithmic Aspects of Wireless Sensor Networks*, 2004, pp. 32–44.

[15] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *IPSN*, 2004, pp. 46–54.

[16] Z. Wang, S. Zheng, S. Boyd, and Y. Ye, "Further relaxations of the SDP approach to sensor network localization," *SIAM J. Optim*, vol. 19, no. 2, pp. 655–673, 2008.

[17] Z. Zhu, A. M.-C. So, and Y. Ye, "Universal Rigidity: Towards Accurate and Efficient Localization of Wireless Networks," in *INFOCOM*, 2010.

[18] P. Biswas, T.-C. Liang, K.-C. Toh, T.-C. Wang, and Y. Ye, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 4, pp. 360–371, 2006.

[19] K. Q. Weinberger, F. Sha, Q. Zhu, and L. Saul, "Graph Laplacian Regularization for Large-Scale Semidefinite Programming," *Advances in Neural Information Processing Systems 19*, pp. 1489–1496, 2007.

[20] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999, http://sedumi.ie.lehigh.edu/.

[21] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, May 2010.

[22] K. Weinberger, http://www.cse.wustl.edu/~kilian/page3/files/lmvu.zip.

[23] D. Goldenberg, A. Krishnamurthy, W. Maness, Y. Yang, A. Young, A. Morse, and A. Savvides, "Network localization in partially localizable networks," in *INFOCOM*, 2005, pp. 313–326.

[24] G. Laman, "On graphs and rigidity of plane skeletal structures," *Journal of Engineering Mathematics*, vol. 4, no. 4, pp. 331–340, 1970.

[25] H. Gluck, "Almost all simply connected closed surfaces are rigid," *Geometric topology*, 1975.

[26] B. Jackson and T. Jordan, "Connected rigidity matroids and unique realizations of graphs," *Journal of Combinatorial Theory Series B*, vol. 94, no. 1, pp. 1–29, 2005.

[27] R. Connelly, "Generic Global Rigidity," *Discrete & Computational Geometry*, vol. 33, no. 4, pp. 549–563, 2005.

[28] S. Gortler, A. Healy, and D. Thurston, "Characterizing generic global rigidity," *arXiv*, vol. 710, 2008. [Online]. Available: http://arxiv.org/pdf/0710.0926

[29] T. Eren, W. Whiteley, and P. N. Belhumeur, "Further results on sensor network localization using rigidity," *in Proceedings of European Workshop on Sensor Networks*, 2005.

[30] Z. Yang and Y. Liu, "Understanding Node Localizability of Wireless Ad-hoc Networks," in *INFOCOM*, 2010.

[31] L. Jian, Z. Yang, and Y. Liu, "Beyond Triangle Inequality: Sifting Noisy and Outlier Distance Measurements for Localization," in *INFOCOM*, 2010.

[32] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *ACM SenSys*, 2004.

[33] J. Liu and Y. Zhang, "Error control in distributed node self-localization," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 162587, 2008.